Document Number: DSP1041

Date: 2009-06-25

Version: 1.1.0

Resource Allocation Profile

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

Copyright Notice

Copyright © 2007, 2009 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a patent owner for such implementations.

For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent may relate to or impact implementations of DMTF standards, visit http://www.dmtf.org/about/policies/disclosures.php.

CONTENTS

Figures

Tables

173

Foreword

The *Resource Allocation Profile* (DSP1041) was prepared by the System Virtualization, Partitioning, and Clustering Workgroup of the DMTF.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgments

The SVPC work group acknowledges the following people for their contributions to the development this profile.

Editors:

- Michael Johanssen – IBM
- Lawrence Lamers – VMware Inc.

Contributors:

- Gareth Bestor – IBM
- Ron Goering – IBM
- Daniel Hiltgen –VMware
- Ron Doyle – IBM
- Rene Schmidt – VMware Inc.
- Steffen Grarup – VMware Inc.
- Hemal Shah – Broadcom
- Fred Maciel – Hitachi Ltd.
- Lawrence Lamers – VMware Inc.
- Andreas Maier – IBM
- John Parchem – Microsoft Corporation
- George Ericson – EMC
- Oliver Benke – IBM
- John Leung – Intel Corporation
- James Fehlig – Novell
- Nihar Shah – Microsoft Corporation
- Shishir Pardikar – Citrix Systems Inc.
- Stephen Schmidt – IBM
- Mark Hapner – Sun Microsystems
- Dave Barrett – Emulex
- John Suit – Fortisphere
- Jeff Wheeler – Cisco
- Mark Johnson – IBM
- Carl Waldsburger - VMware Inc.

Introduction

The information in this specification should be sufficient for a provider or consumer of this data to identify unambiguously the classes, properties, methods, and values that shall be instantiated and manipulated to represent and manage the allocation of resources to systems and subsystems that are modeled using the DMTF Common Information Model (CIM) core and extended model definitions. This includes the allocation and management of resources in support of virtualization platforms.

The target audience for this specification is implementers who are writing CIM-based providers or consumers of management interfaces that represent the component described in this document.

Resource Allocation Profile

1 Scope

The *Resource Allocation Profile* sets the basic resource allocation pattern for resource pools, allocations, and setting data. It also defines the resource-pool-lifecycle management and relationships.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

DMTF DSP0004, *CIM Infrastructure Specification* 2.5,
http://www.dmtf.org/standards/published_documents/DSP0004_2.5.pdf

DMTF DSP0200, *CIM Operations over HTTP* 1.3,
http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

DMTF DSP0207, *WBEM URI Mapping Specification* 1.0,
http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf

DMTF DSP1001, *Management Profile Specification Usage Guide* 1.0,
http://www.dmtf.org/standards/published_documents/DSP1001_1.0.pdf

DMTF DSP1033, *Profile Registration Profile* 1.0,
http://www.dmtf.org/standards/published_documents/DSP1033_1.0.pdf

DMTF DSP1043, *Allocation Capabilities Profile* 1.0,
http://www.dmtf.org/standards/published_documents/DSP1043_1.0.pdf

DMTF DSP1053, *Base Metrics Profile* 1.0,
http://www.dmtf.org/standards/published_documents/DSP1053_1.0.pdf

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply. For the purposes of this document, the terms and definitions in DSP1033 and DSP1001 also apply.

3.1 Standard Terms

For the purposes of this document, the following standard terms and definitions apply.

3.1.1
can
used for statements of possibility and capability, whether material, physical, or causal

3.1.2
cannot
used for statements of possibility and capability, whether material, physical, or causal

252 **3.1.3**
253 **conditional**
254 indicates requirements to be followed strictly to conform to the document if the specified conditions are
255 met

256 **3.1.4**
257 **implementation**
258 an implementation that is conformant to this profile

259 **3.1.5**
260 **mandatory**
261 indicates requirements to be followed strictly to conform to the document and from which no deviation is
262 permitted

263 **3.1.6**
264 **may**
265 indicates a course of action permissible within the limits of the document

266 **3.1.7**
267 **need not**
268 indicates a course of action permissible within the limits of the document

269 **3.1.8**
270 **optional**
271 indicates a course of action permissible within the limits of the document

272 **3.1.9**
273 **referencing profile**
274 indicates a profile that owns the definition of this class and can include a reference to this profile in its
275 “Referenced Profiles” table

276 **3.1.10**
277 **shall**
278 indicates requirements to be followed strictly to conform to the document and from which no deviation is
279 permitted

280 **3.1.11**
281 **shall not**
282 indicates requirements to be followed strictly to conform to the document and from which no deviation is
283 permitted

284 **3.1.12**
285 **should**
286 indicates that among several possibilities, one is recommended as particularly suitable, without
287 mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

288 **3.1.13**
289 **should not**
290 indicates that a certain possibility or course of action is deprecated but not prohibited

291 **3.1.14**
292 **this profile**
293 a reference to this CIM management profile, DSP1041:1.1.0 *Resource Allocation Profile*

3.1.15
unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

3.2 Resource Allocation Terms

For the purposes of this document, the following terms and definitions related to resource allocation apply.

3.2.1
allocated resource
the result of a resource allocation request — the assigned, separated, reserved, or shared part of the resource or emulated resource allocated to the consumer based on the resource allocation request

3.2.2
child pool
pool whose resources are backed by other resource pools; consumer of resources from its parent resource pools; contains no host resources, instead draws resources from parent pools through resource allocations

3.2.3
client
an application that exploits facilities specified by this profile

3.2.4
consumer
entity using allocated resources (for example, a virtual system)

3.2.5
current resource allocation setting data
resource allocation setting data that describes an allocated resource; differs from defined resource allocation setting data if the host system supports the dynamic modification of a resource allocation

3.2.6
dedicated virtual resource
virtual resource that has been given exclusive use of one or more host resources (the host resources are not shared with any other consumer)

3.2.7
defined resource allocation setting data
resource allocation setting data that describes a resource allocation request

3.2.8
host resource
device or computing resource contained by the host system that may be allocated with either exclusive or shared access through the host system to provide resources to a resource pool or consumer

3.2.9
host system
scoping system that contains host resources that are subject to resource allocation

3.2.10
primordial resource pool
pool with no parent and that may aggregate host resources

3.2.11
resource allocation
process of assigning, separating, reserving, granting share of, or emulating resources for use by a consumer

3.2.12
resource allocation request
request for resources to be allocated

3.2.13
resource allocation setting data
RASD
CIM_ResourceAllocationSettingData - settings describing resource allocation; used by a host system to manage the allocation of resources and their relationship to host resources, resource pools used for the allocation, or both

3.2.14
resource pool
abstract entity used by a host system for the purpose of allocating and exposing allocated resources to consumers

3.2.15
resource type
generic type categorizing classes of resources (for example, processor, memory, network adapter)

3.2.16
shared virtual resource
virtual resource that has been given the use of host resources that may also be shared with other consumers

3.2.17
simple resource allocation
resource allocation with no logical device representing the allocated resources

3.2.18
virtual computer system
virtual system as applied to a computer system
Other common industry terms for such a system include *virtual machine*, *hosted computer*, *child partition*, *logical partition*, *domain*, *guest*, and *container*.

3.2.19
virtual resource
resource exposed to a consumer as a logical device based on one or more allocated resources

3.2.20
virtual resource allocation
resource allocation with a logical device representing the allocated resources

3.2.21
virtual system
scoping system that contains virtual resources

4 Symbols and Abbreviated Terms

The following abbreviation is used in this document.

4.1
CIM
Common Information Model

4.2
RASD
CIM_ResourceAllocationSettingData

5 Synopsis

Profile Name: Resource Allocation

Version: 1.1.0

Organization: DMTF

CIM schema version: 2.22

Central Class: CIM_ResourcePool

Scoping Class: CIM_System

The *Resource Allocation Profile* is an abstract profile that extends the management capability of referencing profiles by adding the capability to represent the allocation of resources to consumers. This includes allocation of underlying supporting resources, such as power and cooling, and the allocation of computing resources, such as processors and memory. The resources may be virtualized. A general model is defined by this profile. Requirements and constraints specific to a resource type are defined in a referencing profile dedicated to the resource type. This profile defines a resource pool, allocated resources, allocation settings, and host resources.

This profile shall not be directly implemented. Implementation shall be based on a profile that specializes the requirements of this profile.

Table 1 identifies the profile on which this profile has a dependency.

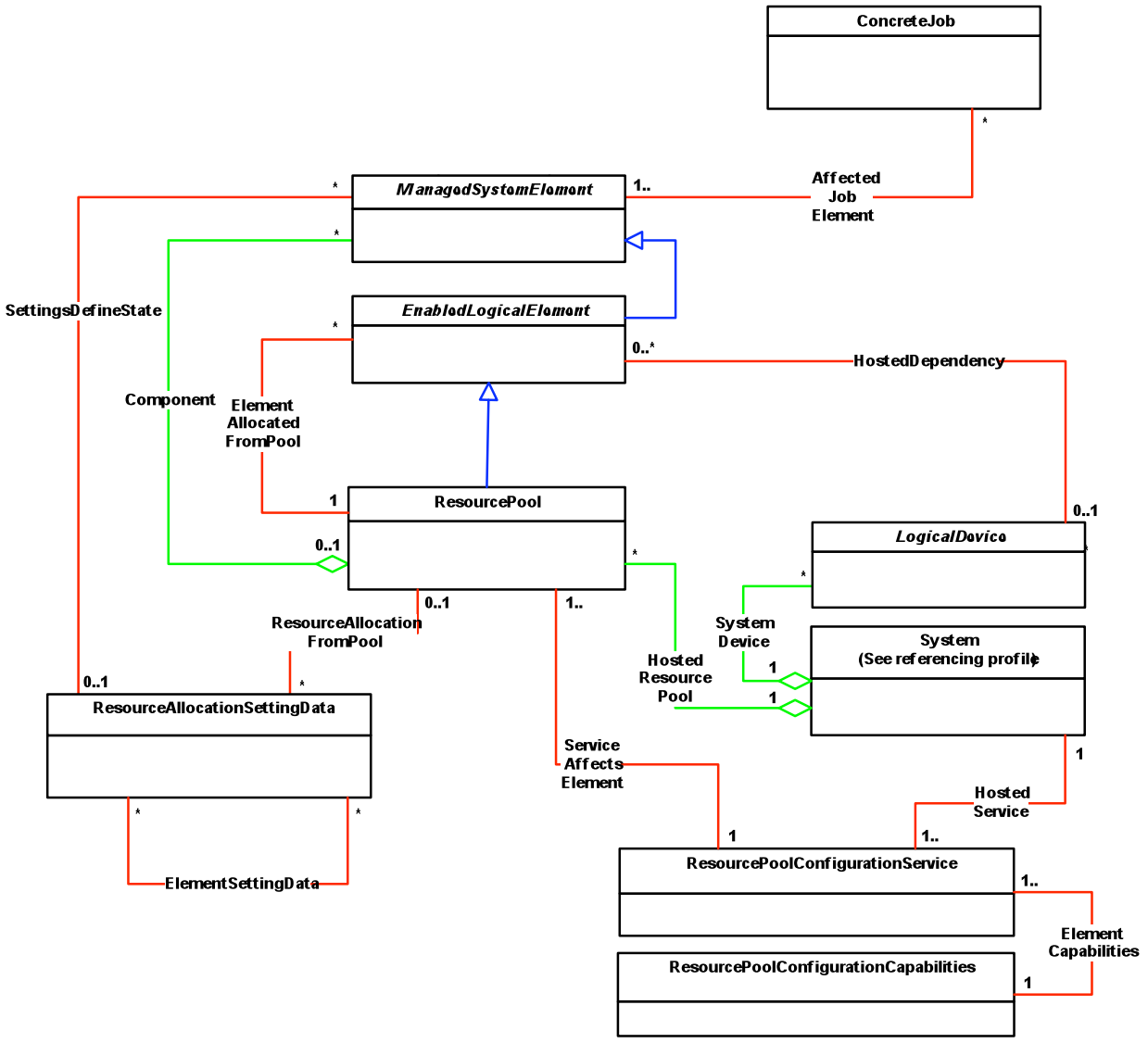
Table 1 – Related Profiles

Profile Name	Organization	Version	Requirement	Description
Allocation Capabilities	DMTF	1.0	Optional	Profile that describes allocation capabilities

6 Description

This section provides an informative description of the management domain addressed by this profile and describes how the CIM elements defined in the profile apply to the management domain.

Figure 1 is the class diagram for the *Resource Allocation Profile*. Cardinalities shown in the diagram reflect the constraints relative to implementations of this profile. For simplicity, the prefix *CIM_* has been removed from the names of the classes.



409

410 **Figure 1 – Resource Allocation Profile: Class Diagram**

411 6.1 General Resource Allocation Concepts

412 The *Resource Allocation Profile* captures the general concept of defining the availability of a given
413 resource type for allocation to consumers. The amount of resource available and the amount of resource
414 allocated are modeled and managed. The aggregation of the underlying components that provide the
415 resource (host resources) may be represented.

416 6.1.1 Host Resources

417 Host resources are those that compose or enable a computer system. Examples include processors,
418 memory, I/O, power supplied to the system, cooling allocated to the system, and so on.

6.1.2 Resource Pool

This profile uses a resource pool as the focal point for resource allocations. Consumers receive resource allocations from resource pools based on resource allocation requests. A resource pool results from aggregation of host resources of a specific type into a resource pool.

For example, when modeling virtualized computer systems, host processors are aggregated into a pool, giving it a known computing capacity, from which virtual processors are allocated. When modeling virtualized storage, the pool may map to a RAID volume or other storage-aggregating construct. Using power as an example, the host power supplies may be aggregated into a pool, thus establishing the total power available for allocation. Power is then allocated to dependent resource consumers from the pool.

6.1.3 Resource Allocation

A resource allocation is a resource that is allocated from a resource pool. A resource allocation request is a request for a resource allocation. A resource allocation is obtained based on a corresponding resource allocation request. Both resource allocation and resource allocation request are represented through instances of the CIM_ResourceAllocationSettingData class.

6.1.4 Hierarchies of Resource Pools

A hierarchy of resource pools may be supported. A hierarchy may be used to provide administrative controls over the set of resources or to partition resources into disjoint sets.

For example, the aggregate processor capability of the computer system may be divided into child pools for individual departments or users.

A hierarchy of resource pools represents the same type of resource and is acyclic. Hierarchies of resource pools consisting of different types of resources are not defined by this profile.

6.1.5 Pool and Resource Management

The creation, deletion, and management of resource pools and assignment of the host resources that they contain are covered by this profile. The allocation of resources to a consumer is covered in derived profiles. The CIM_ResourcePoolConfigurationService class provides extrinsic methods for the management of resource pools.

6.2 Simple Resource Allocation

Simple resource allocation is the allocation of resources (for example, power, cooling, and so on) to a consumer where no logical device represents the resource allocated.

6.3 Virtual Resource Allocation

Virtual resource allocation is the allocation of resources (for example, processor, memory, and so on) to a consumer using an allocated resource. Virtual resource allocation extends the concept of resource allocation with the addition of semantics specific to virtualization. The virtual resources represent the consumer’s view of the allocated resource, which enables management of the allocated resource in the context of management of the consumer. Additional functionality, such as the management of resource allocation definition or state, is introduced.

Multiple virtual resource allocations or virtual resource allocation requests may affect a single resource. For example, a virtual disk may be affected by the allocation of a storage extent and of bandwidth. A resource allocation or resource allocation request may affect one resource or a comprehensive set of resources. For example, a set of virtual processors may be modeled by one resource allocation request.

6.3.1 Virtual Resource

A virtual resource represents the consumer's view of the allocated resource. In a processor example, the virtual processor represents the virtual resource and may be consumed by a virtual computer system. In a storage example, a storage volume represents the virtual resource and may be consumed by a physical or a virtual computer system.

The same CIM class is used to model both the host resource and the virtual resource. In many systems, virtualization-specific details are hidden from the consumer, and the consumer is presented with a virtual resource that looks no different than a host resource (that is, a resource that is not virtualized). The property values contained in the instance of a virtual resource reflect what the consumer of that resource is presented, not the virtualization-specific details of the allocation.

Information specific to resource allocation is modeled using resource allocations and resource allocation requests. These are represented by instances of the CIM_ResourceAllocationSettingData class. This approach ensures that general-purpose management applications may consume information about the virtual resources without having to know details of the underlying virtualization. This approach also prevents proliferation of virtualization-specific properties throughout the CIM schema, or mandatory subclassing of every possible device that may be virtualized. For example, the virtualized processor in the virtual computer system uses the same properties as a physical processor in a physical computer system.

Resource allocation CIM management profiles that specialize this profile may allow or require different CIM classes to be used to represent the virtual and host resources. For example, a consumable storage device may be modeled using CIM_StorageVolume or CIM_LogicalDisk, while the hosting resource may be modeled using CIM_StorageExtent.

6.3.1.1 Dedicated Virtual Resources

Dedicated virtual resources are allocated to a consumer and are not available to other consumers. The host resource backs the virtual resource through a one-to-one mapping that is identified by the MappingBehavior property of CIM_ ResourceAllocationSettingData set to a value of 2 (Dedicated).

6.3.1.2 Shared Virtual Resources

Shared resources may be used by multiple consumers. The host resource may map to multiple virtual resources that may be allocated to the same resource consumer. One host resource is shared by many consumers (for example, a quorum drive in a cluster environment).

A virtual resource may map to different host resources over time. One virtual resource may be mapped to many backing host resources (for example, a virtual processor scheduled to run on different host processors during the course of execution).

6.3.2 Relationship between Host Resource and Virtual Resource

If the virtual resource always maps to the same host resource, the CIM_HostedDependency association may be used to reflect this relationship for a current allocation. Implementations that support scheduling across the pool of host resources transparent to the consumer may not expose the CIM_HostedDependency association because this relationship may change frequently.

6.3.3 Resource Allocation Definition and Resource Allocation State

For each resource that may be allocated, a resource allocation request represents the resource allocation definition. Once one or more elements requested by the resource allocation definition are allocated, a corresponding set of resource allocations represents the resource allocation state. Details of a resource allocation request or a resource allocation are represented by an instance of the CIM_ResourceAllocationSettingData class. Elements of resource allocation definition and resource allocation state may be changed independently.

For example:

- A system may support changing the processor resource allocation for the next boot or reset of the virtual system while the system is running without changing the current state of the system. This behavior is reflected by updates to the defined settings.
- An implementation for virtual memory may support changing the resource allocation definition only. A change such as an increase in virtual memory becomes effective the next time the virtual system is activated.
- An implementation may support changing the settings for the currently running virtual system only where the values revert to the prior settings on the next boot or reset. This behavior is reflected by updates in the current settings only.
- An implementation may support changing the resource allocation definition and state of virtual processors simultaneously. A change such as an increase in the relative share of processing power that a group of processors scoped by a virtual system receive becomes effective immediately without a need for a re-activation, and remains effective beyond the next re-activation.

The kind of changes that are supported are defined in resource allocation CIM management profiles that specialize this profile.

7 Implementation

This section provides normative requirements related to the arrangement of instances and properties of instances for implementations of this profile. The CIM Schema definitions and requirements apply.

7.1 Common Requirements

This section details requirements that shall be met regardless of whether simple resource allocation or virtual resource allocation is implemented. In addition to these common requirements, either the requirements in 7.2 or the requirements in 7.3 shall also be implemented.

7.1.1 Representation of a Resource Pool

Each resource pool managed using this profile shall be represented by an instance of the CIM_ResourcePool class. It shall be associated with the instance of CIM_System that represents the scoping system through one instance of the CIM_HostedResourcePool association. Each resource pool shall represent resources of the same type.

7.1.2 Primordial and Concrete Resource Pools

A resource pool may be primordial or concrete.

Primordial pools aggregate capacity; they represent the known manageable capacity for the host system. Capacity is drawn from the primordial pool to create concrete resource pools or to allocate resources to consumers. There shall be at least one primordial pool for each resource type managed through this profile. The instance of the CIM_ResourcePool class that represents a primordial resource pool shall have the Primordial property set to a value of TRUE.

Concrete resource pools subdivide the resource capacity available at a system. A single concrete pool may represent all the capacity of a primordial pool. The instance of the CIM_ResourcePool class that represents a concrete resource pool shall have the Primordial property set to a value of FALSE.

If a one-to-one correspondence exists between the host resource and the virtual resource, the CIM_HostedDependency association may be used to indicate the correspondence.

7.2 Modeling Virtual Resource Allocation

Virtual resource allocation may be modeled. If virtual resource allocation is modeled, the requirements specified in this section shall be met.

7.2.1 Host Resources

Host resources are modeled as a subclass of the CIM_LogicalDevice class. Host resources may be aggregated into one or more primordial resource pools and allocated to resource pools or resource consumers.

If aggregation of host resources is supported, at least one instance of the CIM_LogicalDevice class shall be associated with the instance of the CIM_ResourcePool class through an instance of the CIM_Component association. If aggregation of host resources is supported, the CIM_ResourceAllocationSettingData.Capacity property shall be supported.

If a resource pool is used for dedicated or shared resources, aggregation of host resources should be supported.

7.2.2 Virtual Resources

Each virtual resource that is fully or partially allocated shall be represented by an instance of the CIM_LogicalDevice class. That instance shall be associated with the instance of the CIM_System class that represents the scoping virtual system through an instance of the CIM_SystemDevice association.

7.2.3 Resource Allocation Definition

Each resource shall have a resource allocation definition. Each element of a resource allocation definition shall be represented by one instance of the CIM_ResourceAllocationSettingData class.

7.2.4 Resource Allocations

An instance of the CIM_LogicalDevice class that represents a virtual resource shall be associated to zero or more instances of the CIM_ResourceAllocationSettingData class that represents the resource allocation state through an instance of the CIM_SettingsDefineState association.

One instance of the CIM_ResourceAllocationSettingData class may be associated with more than one instance of the CIM_LogicalDevice class that represents a set of virtual resources.

An instance of the CIM_ResourceAllocationSettingData class that represents the current allocation state shall be associated with a virtual resource through an instance of the CIM_SettingsDefineState association.

An instance of the CIM_ResourceAllocationSettingData class that represents the defined allocation state shall be associated with the instance of the CIM_ResourceAllocationSettingData class that represents the current allocation state through an instance of the CIM_ElementSettingData association with the IsDefault property set to 1 (Is Default).

The non-key properties of the two instances of the CIM_ResourceAllocationSettingData class may match if both the current and defined settings are the same.

An instance of the CIM_ResourceAllocationSettingData class that represents a current resource allocation shall be associated with one instance of the CIM_ResourcePool class through an instance of the CIM_ResourceAllocationFromPool association.

An instance of CIM_ResourceAllocationSettingData that represents a defined resource allocation shall not be associated with instances of CIM_ResourcePool through the CIM_ResourceAllocationFromPool association.

7.2.5 Dedicated Allocations

If the value of the MappingBehavior property is set to 2 (Dedicated) in the instance of the CIM_ResourceAllocationSettingData class that represents the defined allocation state, and if no values are specified for the HostResource[ ] array property or the HostResource[ ] array property is not specified (NULL), the system shall select the host resources if the virtual resource is allocated.

If the value of the MappingBehavior property is set to 2 (Dedicated) in the instance of the CIM_Resource-AllocationSettingData class that represents the current allocation state, the HostResource[ ] array property shall contain the identities of host resources that are dedicated to the virtual resource. For dedicated resources, an instance of the CIM_HostedDependency association may be present between the instance of the CIM_LogicalDevice class that represents a dedicated host resource allocation and the instance of the CIM_LogicalDevice class that represents the virtual resource.

7.2.6 Allocations with Affinity

Virtual resources may be allocated with affinity to host resources using values of the MappingBehavior property.

If the MappingBehavior property is set to 4 (Hard Affinity), only the resources specified in the HostResource[ ] array property shall be used. If no values are specified for the HostResource[ ] array property or the HostResource[ ] array property is not specified (NULL), the system shall select the host resources if the virtual resource is allocated and maintain the allocation of those resources to the virtual device.

If the MappingBehavior property is set to 3 (Soft Affinity), the resources specified in the HostResource[ ] array property are preferred, but alternative host resources may be used. If no values are specified for the HostResource[ ] array property or the HostResource[ ] array property is not specified (NULL), the system shall select the host resources if the virtual resource is allocated.

If the MappingBehavior property is set to 3 (Soft Affinity) or 4 (Hard Affinity), an instance of the CIM_HostedDependency association shall not be used between the instance of the CIM_LogicalDevice class that represents a dedicated host resource allocation and the instance of the CIM_LogicalDevice class that represents the virtual resource.

If values are specified for the HostResource[ ] array property, the number of resources listed in the HostResource[ ] array property shall be adequate to satisfy the allocation request but may include additional resources.

The HostResource[ ] array property that represents the defined allocation state shall be set to the user's request. The HostResource[ ] array property that represents the current allocation state shall be set to the current active behavior.

7.3 Modeling Simple Resource Allocation

Simple resource allocation may be modeled. If simple resource allocation is modeled, the requirements in this section shall be met.

7.3.1 General Requirements

Each instance of the CIM_ResourceAllocationSettingData class that represents a current allocation state or alternate allocation state shall be associated with one instance of the CIM_ResourcePool class through an instance of the CIM_ResourceAllocationFromPool association.

A logical device shall not be instantiated.

7.3.2 Current Allocation

An instance of the CIM_ResourceAllocationSettingData class that represents the current state shall be associated with one instance of the CIM_ResourceAllocationSettingData class through an instance of the CIM_ElementSettingData association with the IsCurrent property set to 1 (Is Current). If the CIM_ResourceAllocationSettingData class that represents the current state is modified, the IsCurrent property shall be set to a value other than 1 (Is Current).

7.3.3 Alternate Allocations

Alternate allocations of the resource for the consumer may be supported. Each alternate allocation state shall be represented by an instance of the CIM_ResourceAllocationSettingData class that is associated with an instance of the CIM_ResourceAllocationSettingData class that represents the alternate allocation state through an instance of the CIM_ElementSettingData association with the IsCurrent property set to 2 (Is Not Current).

7.4 Resource Pool Management

Resource pool management may be modeled. If resource pool management is modeled, the requirements of this section shall be met. Implementations may support active management of instances of the CIM_ResourcePool class, or they may expose a read-only view of existing instances of the CIM_ResourcePool class.

An instance of the CIM_ResourcePoolConfigurationService class shall be implemented; however, the methods of the service are optional. The instance of the CIM_ResourcePoolConfigurationService class shall be associated with the host system through an instance of the CIM_HostedService association. One instance of CIM_ResourcePoolConfigurationCapabilities shall be associated with the CIM_ResourcePoolConfigurationService instance through the CIM_ElementCapabilities association. This instance of CIM_ResourcePoolConfigurationCapabilities shall reflect the methods supported. If active management is not supported by an implementation, all properties of the associated CIM_ResourcePoolConfigurationCapabilities instance shall be set to NULL.

7.5 Metrics

If metrics are implemented, the DSP1053 shall be implemented. If the instance of the CIM_BaseMetricDefinition class defines a metric that applies across the entire resource pool, the instance of CIM_BaseMetricDefinition class shall be associated with an instance of the CIM_ResourcePool class through the CIM_MetricDefForME association, and the instance of the CIM_BaseMetricDefinition class shall not be associated with any other instances of the CIM_ManagedElement class. An example of this type of metric is a metric that reports the total instantaneous resource consumption from the pool.

If the instance of the CIM_BaseMetricDefinition class defines a metric related to an individual virtual device's utilization of resources from the resource pool, the instance of the CIM_BaseMetricDefinition class shall be associated with the instance of the CIM_ResourcePool class through the CIM_MetricDefForME association, and the instance of the CIM_BaseMetricDefinition class shall be associated with the instance of the CIM_LogicalDevice class that represents the virtual device through an instance of the CIM_MetricDefForME association.

If the instance of the CIM_BaseMetricDefinition class defines a metric for the virtual device that is not related to the consumption by the device of resources from the resource pool, the instance of the CIM_BaseMetricDefinition class shall not be associated with the instance of the CIM_ResourcePool class.

If the instance of the CIM_BaseMetricDefinition class defines a metric related to the resource pool and a host resource, the instance of the CIM_BaseMetricDefinition class shall be associated with the instance of the CIM_ResourcePool class through an instance of the CIM_MetricDefForME association, and the

672 instance of the CIM_BaseMetricDefinition class shall be associated with the instance of the
673 CIM_ManagedElement class that represents the host resource through an instance of the
674 CIM_MetricDefForME association.

675 7.6 Resource Pool Hierarchies

676 Hierarchies of resource pools may be modeled. A hierarchy of resource pools represents the same type
677 of resource and shall be acyclic.

678 Child pools may be allocated from the parent pool using the CIM_ResourcePoolConfigurationService
679 class.

680 Parent and child pools shall be scoped to the same system.

681 A pool may have virtual resources and child pools allocated from it simultaneously.

682 7.7 Virtual Resource Definition and Modification

683 The CIM_ResourceAllocationSettingData class is used as an input for virtual system definition. The client
684 and implementation considerations are defined. The *Resource Allocation Profile* specifies how to define
685 and modify virtual resources using methods of the virtual system management service. In these method
686 specifications, the CIM_ResourceAllocationSettingData class is used for parameterization of resource-
687 allocation-specific properties. The capabilities model may be used to convey information about limitations
688 for and default values of properties of the CIM_ResourceAllocationSettingData class; see DSP1043
689 clause 9.

690 7.7.1 CIM_ResourceAllocationSettingData.InstanceID Property

691 A client shall set the InstanceID property to NULL if the instance of the
692 CIM_ResourceAllocationSettingData class is created locally. A client shall not modify the InstanceID
693 property in an instance of the CIM_ResourceAllocationSettingData class that was received from an
694 implementation and is sent back to the implementation as a parameter of a modification method.

695 An implementation shall ignore any non-NULL value in a definition request.

696 In a modification request, an implementation shall use a non-NULL value to identify an existing instance
697 of the CIM_ResourceAllocationSettingData class. If a value is specified that does not identify an instance
698 of the CIM_ResourceAllocationSettingData class, an implementation shall return a return code that
699 indicates an invalid parameter; see Clause 8.

700 7.7.2 CIM_ResourceAllocationSettingData.ResourceType Property

701 A client shall set the value of the ResourceType property to designate the type of the virtual resource
702 allocation request.

703 The implementation shall use the value of the ResourceType property, as well as the value of the
704 OtherResourceType property if the value of the ResourceType property is 1 (Other), to determine the
705 type of the virtual resource allocation request. If the implementation does not support the requested
706 resource type, it shall fail the method execution.

707 7.7.3 CIM_ResourceAllocationSettingData.OtherResourceType Property (Conditional)

708 If a client sets the value of the ResourceType property to a value other than 1 (Other), it shall set the
709 value of OtherResourceType property to NULL. If a client sets the value of the ResourceType property to
710 1 (Other), it shall set the value of the OtherResourceType property to identify the type of the virtual
711 resource allocation request in an implementation-dependent way.

712 The implementation shall use the value of the OtherResourceType property if the value of the
713 ResourceType property is 1 (Other) to determine the type of the virtual resource allocation request. If the
714 implementation does not support the requested resource type, it shall fail the method execution.

715 7.7.4 CIM_ResourceAllocationSettingData.ResourceSubType Property

716 A client may set the value of the ResourceSubType property to designate the subtype of the virtual
717 resource allocation request. A client may set the value of the ResourceSubType property to NULL,
718 requesting default behavior.

719 The implementation shall use the value of the ResourceSubType property to determine the subtype of the
720 virtual resource allocation request. If the implementation does not support the requested virtual resource
721 subtype, it shall return a return code that indicates an invalid parameter; see Clause 8.

722 7.7.5 CIM_ResourceAllocationSettingData.PoolID Property

723 A client may set the value of the PoolID property to designate a resource pool that shall be used for
724 resource allocation. In this case, the values of the PoolID and ResourceType properties shall be equal to
725 the values of respective properties in an instance of the CIM_ResourcePool class that represents the
726 designated resource pool. A client may set the value of the PoolID property to NULL, requesting default
727 behavior.

728 An implementation shall use the value of the PoolID and ResourceType properties to assign the resource
729 pool that shall be used for the resource allocation. If the value of the PoolID property is NULL, the
730 implementation may assign a default resource pool. If no resource pool exists with matching values of the
731 PoolID and ResourceType properties, the implementation may either assign a default resource pool or fail
732 the method execution. An implementation may defer the selection of a default resource pool until
733 resource allocation actually occurs.

734 7.7.6 CIM_ResourceAllocationSettingData.ConsumerVisibility Property

735 A client may set the value of the ConsumerVisibility property to specify whether the virtual resource or
736 comprehensive set of virtual resources that is requested by the virtual resource allocation request shall be
737 virtualized or shall be one or more passed-through host resources. A client may specify a value of NULL
738 for the ConsumerVisibility property, requesting a default behavior.

739 If the property is set to a value other than NULL, the client shall perform one of the following actions:

740 • set a value of 0 (Unknown) to request default behavior (with the same effect as a value of
741 NULL)

742 • set a value of 2 (Passed-Through) to specify that one or more passed-through host devices
743 shall be allocated to the virtual resource requested by this virtual resource allocation request,
744 and shall provide one or more elements in the HostResource[ ] array property that identify the
745 host resources that shall be passed through

746 • set a value of 3 (Virtualized) to specify that the virtual resource that results from this virtual re-
747 source allocation request shall be virtualized

748 The client shall not use a value of 4 (Not Represented).

749 An implementation shall use the value of the ConsumerVisibility property to determine whether the virtual
750 resource or comprehensive set of virtual resources requested by this virtual resource allocation request
751 shall be virtualized or shall be a passed-through host resource.

752 • If the value of the ConsumerVisibility property is NULL or 0 (Unknown), the implementation may
753 exhibit an implementation-specific default behavior that may also depend on the resource type,
754 the selected resource pool, or both.

- If the value of the ConsumerVisibility property is 2 (Passed-Through), the implementation shall establish a virtual resource allocation request or virtual resource allocation for host resources as specified by elements of the HostResource[ ] array property as passed-through devices in the resulting virtual resource allocation. If no values are specified by the HostResource[ ] array property, the implementation may exhibit an implementation-specific default behavior.
- If the value of the ConsumerVisibility property is 3 (Virtualized), the implementation shall establish a virtual resource allocation request or virtual resource allocation for a virtualized virtual device or a comprehensive set of virtualized virtual devices.
- If the value of the ConsumerVisibility property is 4 (Not Represented), the implementation shall fail the method execution.

7.7.7 CIM_ResourceAllocationSettingData.HostResource[ ] Array Property

A client may set the value of the HostResource[ ] array property to indicate that the requested virtual resource allocation shall be based on host resources that are identified by element values. The kind of dependency is specified through the ConsumerVisibility property (see 7.7.6) and the MappingBehavior property (see 7.7.18). A client may set the value of the HostResource[ ] array property to NULL or may specify an empty array in order to request the implementation to decide whether the requested resource allocation shall be directly based on host resources.

An implementation shall use the value of the HostResource[ ] array property to determine whether and how the requested virtual resource allocation shall be based on host resources. Respective host resources are identified by element values of the HostResource[ ] array property. The implementation shall use the value of other properties such as ConsumerVisibility (see 7.7.6) and MappingBehavior (see 7.7.18) to determine the kind of dependency. If no value or an empty array is provided as the value of the HostResource[ ] array property, the implementation may provide a pure virtual resource for the request or may select host resources at its own discretion.

7.7.8 CIM_ResourceAllocationSettingData.AllocationUnits Property

A client should set the value of the AllocationUnits property to specify a unit of measurement for the virtual resource allocation request. The unit of measurement shall be compatible with the requested resource type. A client may set the value of the AllocationUnits property to NULL, requesting the implementation to assume a resource-type-specific default value for the unit of measurement.

An implementation shall use the value of the AllocationUnits property to determine the unit of measurement for the virtual resource allocation request. If the provided value is not compatible with the resource type, the implementation shall fail the method execution. If a value is not provided (NULL), the implementation shall assume a resource-type-specific default value for the unit of measurement. A resource-type-specific resource allocation DMTF management profile may specify rules for the determination of the default value.

7.7.9 CIM_ResourceAllocationSettingData.VirtualQuantity Property

A client should set the value of the VirtualQuantity property to specify the quantity of virtual resources that shall result from the virtual resource allocation request. A client may set the value of the VirtualQuantity property to NULL, requesting a default behavior. A resource-type-specific resource allocation DMTF management profile may specify rules for the determination of a default value.

7.7.10 CIM_ResourceAllocationSettingData.Reservation Property

A client may set the value of the Reservation property to specify the amount of host resource that is requested by the virtual resource allocation request. The unit of measurement established by the value of the AllocationUnits property applies. A client may set the value of the Reservation property to NULL, requesting a default behavior.

An implementation shall use the value of the Reservation property to determine the amount of host resource that is requested by the virtual resource allocation request. If a value is not provided (NULL), the implementation may exhibit an implementation-specific default behavior. A resource-type-specific resource allocation DMTF management profile may specify rules for the determination of a default value.

7.7.11 CIM_ResourceAllocationSettingData.Limit Property

A client may set the value of the Limit property to specify the maximum amount of host resource that is requested by the virtual resource allocation request. The unit of measurement established by the value of the AllocationUnits property applies. A client may set the value of the Limit property to NULL, requesting a default behavior.

An implementation shall use the value of the Limit property to determine the maximum amount of host resource that is requested by the virtual resource allocation request. If a value is not provided (NULL), the implementation may exhibit an implementation-specific default behavior. A resource-type-specific resource allocation DMTF management profile may specify rules for the determination of a default value.

7.7.12 CIM_ResourceAllocationSettingData.Weight Property

A client may set the value of the Weight property to specify a relative weight that is requested by the virtual resource allocation request with respect to other virtual resource allocation requests from the same resource pool. A client may set the value of the Weight property to NULL, requesting a default behavior.

An implementation shall use the value of the Weight property to determine a relative weight that is requested by the virtual resource allocation request with respect to other virtual resource allocation requests from the same resource pool. If a value is not provided (NULL), the implementation may exhibit an implementation-specific default behavior. A resource-type-specific resource allocation DMTF management profile may specify rules for the determination of a default value.

7.7.13 CIM_ResourceAllocationSettingData.AutomaticAllocation Property

A client may set the value of the AutomaticAllocation property to specify that the requested resource allocation is obtained automatically when the virtual system is activated. A client may set the value of the AutomaticAllocation property to NULL, requesting a default value of TRUE.

An implementation shall use the value of the AutomaticAllocation property to determine whether the requested resource allocation is obtained automatically when the virtual system is activated. The default value shall be TRUE, requesting automatic resource allocation. Resource-type-specific resource allocation DMTF management profiles may specify a different default behavior.

7.7.14 CIM_ResourceAllocationSettingData.AutomaticDeallocation Property

A client may set the value of the AutomaticDeallocation property to specify that the requested resource allocation is released automatically when the virtual system is deactivated. A client may set the value of the AutomaticDeallocation property to NULL, requesting a default value of TRUE.

An implementation shall use the value of the AutomaticDeallocation property to determine whether the resource allocation is automatically released when the virtual system is de-activated. The default value shall be TRUE, requesting automatic resource deallocation. Resource-type-specific resource allocation DMTF management profiles may specify a different default behavior.

7.7.15 CIM_ResourceAllocationSettingData.Parent Property

A client may set the value of the Parent property to specify a parent resource required to establish the resource allocation. An example of such a parent resource would be a controller. A client may set the value of the Parent property to NULL, requesting a default behavior.

An implementation shall use the value of the Parent property to determine if and which parent resource is required for the requested resource allocation. If no value is specified (NULL), the implementation may decide whether a parent resource is needed and eventually select one itself. If a value is specified, the implementation shall base the resource allocation request on the requested parent resource. If the requested parent resource is not capable to support the requested resource allocation, the implementation shall fail the request.

7.7.16 CIM_ResourceAllocationSettingData.Connection[ ] Array Property

A client may set the value of the Connection[ ] array property to specify connection information for the requested resource allocation. An example of connection information is the target network for a network adaptor or the target switch port for storage adaptors. Connection information is highly dependent on resource type and implementation; for details, refer to the resource-type-specific resource allocation DMTF management profile.

An implementation shall use the values within the Connection[ ] array property to determine connection information for the requested resource allocation. If no value is specified (NULL), the implementation may decide whether the requested resource allocation requires connection information and establish a default connection.

7.7.17 CIM_ResourceAllocationSettingData.Address Property

A client may set the value of the Address property to specify an address for the new virtual device. In general, the requirement for the value of the Address property will depend on the resource type. For a particular resource type, restrictions on the potential value set may exist.

An implementation shall interpret the value of the Address property such that the new virtual resource adopts that address value while it is instantiated. If no value is specified (NULL), the implementation may assign a value for the Address that is specific to the implementation and resource type.

7.7.18 CIM_ResourceAllocationSettingData.MappingBehavior Property

A client may set the value of the MappingBehavior property to specify whether the requested resource allocation has an affinity to or is directly based on host resources that are specified in the optional HostResource[ ] array property (see 7.7.7). A client may set the value of the MappingBehavior property to NULL or to 0 (Unknown) to request that the implementation shall decide on the mapping behavior. A client shall not specify a value of 1 (Not Supported).

An implementation shall use the value of the MappingBehavior property to determine how the requested resource allocation depends on host resources that are specified in the HostResource[ ] array property.

- If the value is 1 (Not Supported), the implementation shall return a return code that indicates an invalid parameter; see Clause 8.
- If the value is not provided (NULL), is 0 (Unknown), or is 1 (Not Supported), the implementation shall establish an implementation-specific default behavior. The resource request may or may not be mapped to or based on host resources depending on the implementation’s decision.
- If the value is 2 (Dedicated), the implementation shall establish a direct mapping of the virtual resource onto the resources specified through the HostResource[ ] array property. The implementation may establish a mapping in the resource allocation request resulting from this instance of the CIM_ResourceAllocationSettingData class; however, it is possible that the requested resources are not available at resource allocation time, resulting in an error condition at that time.

8 Methods

This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM elements defined by this profile.

8.1 CIM_ResourcePoolConfigurationService.CreateChildResourcePool( )

The CIM Schema description of this method applies. This optional method creates (or starts a job to create) a nested resource pool. Refer to the MOF for a detailed description.

If the SupportedSyncMethods property of the associated instance of CIM_ResourcePoolConfigurationCapabilities is set to 3 (CreateChildResourcePool Is Supported), the CreateChildResourcePool( ) method shall be implemented and shall not return a value of 1 or 4096.

If the SupportedAsyncMethods property of the associated instance of CIM_ResourcePoolConfigurationCapabilities is set to 3 (CreateChildResourcePool Is Supported), the CreateChildResourcePool( ) method shall be implemented and shall not return a value of 1.

If neither the SupportedSyncMethods nor the SupportedAsyncMethods property of the associated instance of CIM_ResourcePoolConfigurationCapabilities is set to 3 (CreateChildResourcePool Is Supported), the CreateChildResourcePool( ) method shall not be implemented, or if the method is implemented, it shall return the value 1.

If an implementation supports the resource pool hierarchy device model, it uses this method to create child pools.

Detailed requirements of the CreateChildResourcePool( ) method are specified in Table 2 and Table 3.

No standard messages are defined.

Table 2 – CIM_ResourcePoolConfigurationService.CreateChildResourcePool( ) Method: Return Code Values

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
6	In use
7	Incorrect ResourceType for the pool
8	Insufficient resources
4096	Method parameters checked – job started

906 **Table 3 – CIM_ResourcePoolConfigurationService.CreateChildResourcePool( ) Method:**
907 **Parameters**

Qualifiers	Name	Type	Description/Values
IN	ElementName	String	The desired name of the resource pool
IN	Settings	String	A string representation of a CIM_ResourceAllocationSettingData instance that represents the allocation assigned to this child pool
IN	ParentPool	CIM_ResourcePool REF	The parent pool from which to create this pool
OUT	Pool	CIM_ResourcePool REF	The resulting resource pool
OUT	Job	CIM_ConcreteJob REF	Returned job if started
OUT	Error	String	Encoded error instance if the operation failed and did not return a job

908 ## 8.2 CIM_ResourcePoolConfigurationService.DeleteResourcePool( )

909 The CIM Schema description of this method applies. This optional method deletes (or starts a job to
910 delete) a resource pool. Refer to the MOF for a detailed description.

911 If the SupportedSyncMethods property of the associated instance of
912 CIM_ResourcePoolConfigurationCapabilities is set to 4 (DeleteResourcePool Is Supported), the
913 DeleteResourcePool( ) method shall be implemented and shall not return a value of 1 or 4096.

914 If the SupportedAsyncMethods property of the associated instance of
915 CIM_ResourcePoolConfigurationCapabilities is set to 4 (DeleteResourcePool Is Supported), the
916 DeleteResourcePool( ) method shall be implemented and shall not return a value of 1.

917 If neither the SupportedSyncMethods nor the SupportedAsyncMethods property of the associated
918 instance of CIM_ResourcePoolConfigurationCapabilities is set to 4 (DeleteResourcePool Is Supported),
919 the DeleteResourcePool( ) method shall not be implemented, or if the method is implemented, it shall
920 return the value 1.

921 Detailed requirements of the DeleteResourcePool( ) method are specified in Table 4 and Table 5.

922 No standard messages are defined.

923 **Table 4 – CIM_ResourcePoolConfigurationService.DeleteResourcePool( ) Method: Return Code**
924 **Values**

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
6	In use
7	Incorrect ResourceType for the pool
4096	Method parameters checked – job started

925 **Table 5 – CIM_ResourcePoolConfigurationService.DeleteResourcePool( ) Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	Pool	CIM_ResourcePool REF	The resource pool to delete
OUT	Job	CIM_ConcreteJob REF	Returned job if started
OUT	Error	String	Encoded error instance if the operation failed and did not return a job

926 ## 8.3 CIM_ResourcePoolConfigurationService.AddResourcesToResourcePool( )

927 The CIM Schema description of this method applies. This optional method adds (or starts a job to add)
928 resources to a resource pool. Refer to the MOF for a detailed description.

929 If the SupportedSyncMethods property of the associated instance of
930 CIM_ResourcePoolConfigurationCapabilities is set to 5 (AddResourcesToResourcePool Is Supported),
931 the AddResourcesToResourcePool( ) method shall be implemented and shall not return a value of 1 or
932 4096.

933 If the SupportedAsyncMethods property of the associated instance of
934 CIM_ResourcePoolConfigurationCapabilities is set to 5 (AddResourcesToResourcePool Is Supported),
935 the AddResourcesToResourcePool( ) method shall be implemented and shall not return a value of 1.

936 If neither the SupportedSyncMethods nor the SupportedAsyncMethods property of the associated
937 instance of CIM_ResourcePoolConfigurationCapabilities is set to 5 (AddResourcesToResourcePool Is
938 Supported), the AddResourcesToResourcePool( ) method shall not be implemented, or if the method is
939 implemented, it shall return the value 1.

940 Detailed requirements of the AddResourcesToResourcePool( ) method are specified in Table 6 and
941 Table 7.

942 No standard messages are defined.

943 **Table 6 – CIM_ResourcePoolConfigurationService.AddResourcesToResourcePool( ) Method:**
944 **Return Code Values**

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
6	In use
7	Incorrect ResourceType for the pool
4096	Method parameters checked – job started

945 **Table 7 – CIM_ResourcePoolConfigurationService.AddResourcesToResourcePool( ) Method:**
946 **Parameters**

Qualifiers	Name	Type	Description/Values
IN	HostResource[ ]	CIM_LogicalDevice REF[ ]	The host resources to assign to the pool
IN	Pool	CIM_ResourcePool REF	The primordial ResourcePool to add resources to
OUT	Job	CIM_ConcreteJob REF	Returned job if started
OUT	Error	String	Encoded error instance if the operation failed and did not return a job

947 ## 8.4 CIM_ResourcePoolConfigurationService.RemoveResourcesFromResourcePool( )

948 The CIM Schema description of this method applies. This optional method removes (or starts a job to
949 remove) resources from a resource pool. Refer to the MOF for a detailed description.

950 If the SynchronousMethodsSupported property of the associated instance of
951 CIM_ResourcePoolConfigurationCapabilities is set to 6 (RemoveResourcesFromResourcePool Is
952 Supported), the RemoveResourcesFromResourcePool( ) method shall be implemented and shall not
953 return a value of 1 or 4096.

954 If the AsynchronousMethodsSupported property of the associated instance of
955 CIM_ResourcePoolConfigurationCapabilities is set to 6 (RemoveResourcesFromResourcePool Is
956 Supported), the RemoveResourcesFromResourcePool( ) method shall be implemented and shall not
957 return a value of 1.

958 If neither the SynchronousMethodsSupported nor the AsynchronousMethodsSupported property of the
959 associated instance of CIM_ResourcePoolConfigurationCapabilities is set to 6
960 (RemoveResourcesFromResourcePool Is Supported), the RemoveResourcesFromResourcePool( )
961 method shall not be implemented, or if the method is implemented, it shall return the value 1.

962 Detailed requirements of the RemoveResourcesFromResourcePool( ) method are specified in Table 8
963 and Table 9.

964 No standard messages are defined.

965 **Table 8 – CIM_ResourcePoolConfigurationService.RemoveResourcesFromResourcePool( )**
966 **Method: Return Code Values**

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
6	In use
7	Incorrect ResourceType for the pool
8	Insufficient resources
4096	Method parameters checked – job started

967 **Table 9 – CIM_ResourcePoolConfigurationService.RemoveResourcesFromResourcePool( )**
968 **Method: Parameters**

Qualifiers	Name	Type	Description/Values
IN	HostResource[ ]	CIM_LogicalDevice REF[ ]	The host resources to remove from the pool
IN	Pool	CIM_ResourcePool REF	The primordial ResourcePool to remove resources from
OUT	Job	CIM_ConcreteJob REF	Returned job if started
OUT	Error	String	Encoded error instance if the operation failed and did not return a job

969 ## 8.5 CIM_ResourcePoolConfigurationService.ChangeParentResourcePool( )

970 The CIM Schema description of this method applies. This optional method changes (or starts a job to
971 change) a parent resource pool. Refer to the MOF for a detailed description.

972 If the SupportedSyncMethods property of the associated instance of
973 CIM_ResourcePoolConfigurationCapabilities is set to 7 (ChangeParentResourcePool Is Supported), the
974 ChangeParentResourcePool( ) method shall be implemented and shall not return a value of 1 or 4096.

975 If the SupportedAsyncMethods property of the associated instance of
976 CIM_ResourcePoolConfigurationCapabilities is set to 7 (ChangeParentResourcePool Is Supported), the
977 ChangeParentResourcePool( ) method shall be implemented and shall not return a value of 1.

978 If neither the SupportedSyncMethods nor the SupportedAsyncMethods property of the associated
979 instance of CIM_ResourcePoolConfigurationCapabilities is set to 7 (ChangeParentResourcePool Is
980 Supported), the ChangeParentResourcePool( ) method shall not be implemented, or if the method is
981 implemented, it shall return the value 1.

982 Detailed requirements of the ChangeParentResourcePool( ) method are specified in Table 10 and
983 Table 11.

984 No standard messages are defined.

985 **Table 10 – CIM_ResourcePoolConfigurationService.ChangeParentResourcePool( ) Method: Return**
986 **Code Values**

Value	Description
0	Job completed with no error
1	Not supported
2	Unknown
3	Timeout
4	Failed
5	Invalid parameter
6	In use
7	Incorrect ResourceType for the pool
8	Insufficient resources
4096	Method parameters checked – job started

987 **Table 11 – CIM_ResourcePoolConfigurationService.ChangeParentResourcePool( ) Method: Parameters**
988

Qualifiers	Name	Type	Description/Values
IN	ParentPool	CIM_ResourcePool REF	The parent resource pool to change to
IN	Settings	String	A string representation of an instance of CIM_ResourceAllocationSettingData that represents the allocation assigned to this child pool
OUT	Job	CIM_ConcreteJob REF	Returned job if started
OUT	Error	String	Encoded error instance if the operation failed and did not return a job

989 ## 8.6 Profile conventions for operations

990 For each profile class (including associations), the implementation requirements for operations, including
991 those in the following default list, are specified in class-specific subclauses of this clause.

992 The default list of operations for all classes is:

993 GetInstance( )

994 EnumerateInstances( )

995 EnumerateInstanceNames( )

996 For classes that are referenced by an association, the default list also includes

997 Associators( )

998 AssociatorNames( )

999 References( )

1000 ReferenceNames( )

1001 ## 8.7 CIM_AffectedJobElement

1002 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1003 NOTE Related profiles may define additional requirements on operations for the profile class.

1004 ## 8.8 CIM_BaseMetricDefinition

1005 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1006 NOTE Related profiles may define additional requirements on operations for the profile class.

1007 ## 8.9 CIM_BaseMetricValue

1008 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1009 NOTE Related profiles may define additional requirements on operations for the profile class.

1010 ## 8.10 CIM_Component

1011 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.11 CIM_ConcreteJob

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.12 CIM_ElementAllocatedFromPool

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.13 CIM_ElementCapabilities

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.14 CIM_ElementSettingData

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.15 CIM_HostedDependency

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.16 CIM_HostedResourcePool

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.17 CIM_HostedService

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.18 CIM_LogicalDevice

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

8.19 CIM_MetricDefForME

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1039 NOTE Related profiles may define additional requirements on operations for the profile class.

1040 ## 8.20 CIM_MetricForME

1041 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1042 NOTE Related profiles may define additional requirements on operations for the profile class.

1043 ## 8.21 CIM_MetricInstance

1044 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1045 NOTE Related profiles may define additional requirements on operations for the profile class.

1046 ## 8.22 CIM_ResourceAllocationFromPool

1047 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1048 NOTE Related profiles may define additional requirements on operations for the profile class.

1049 ## 8.23 CIM_ResourceAllocationSettingData

1050 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1051 NOTE Related profiles may define additional requirements on operations for the profile class.

1052 ## 8.24 CIM_ResourcePool

1053 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1054 NOTE Related profiles may define additional requirements on operations for the profile class.

1055 ## 8.25 CIM_ResourcePoolConfigurationCapabilities

1056 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1057 NOTE Related profiles may define additional requirements on operations for the profile class.

1058 ## 8.26 CIM_ResourcePoolConfigurationService

1059 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1060 NOTE Related profiles may define additional requirements on operations for the profile class.

1061 ## 8.27 CIM_ServiceAffectsElement

1062 All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

1063 NOTE Related profiles may define additional requirements on operations for the profile class.

8.28 CIM_SystemDevice

All operations in the default list in 8.6 shall be implemented as defined in DSP0200.

NOTE Related profiles may define additional requirements on operations for the profile class.

9 Use Cases

This clause contains object diagrams and use cases that represent the intended use of this profile. The use cases are informative and not intended to define the requirements for conformance.

9.1 Abstract Instance Diagram

Figure 2 illustrates the use of the *Resource Allocation Profile* with a primordial pool, a concrete pool, and backed resources used for virtualization.

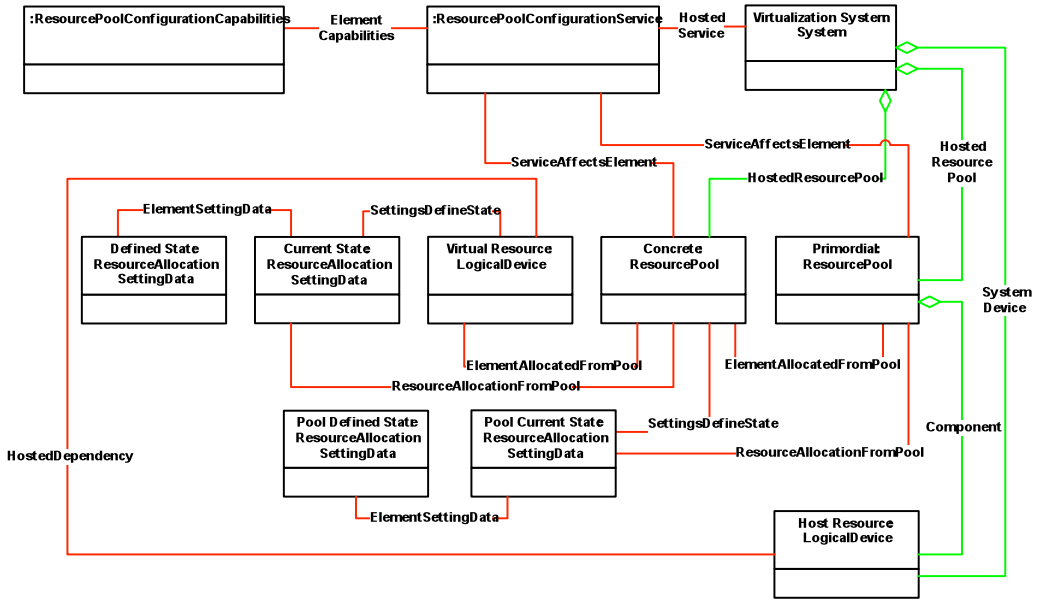


Figure 2 – Abstract Instance Diagram: Concrete Resource Pool

Figure 3 illustrates the use of the *Resource Allocation Profile* with a primordial pool and backed resources used for virtualization. Resources are allocated directly from the primordial pool to consumers.

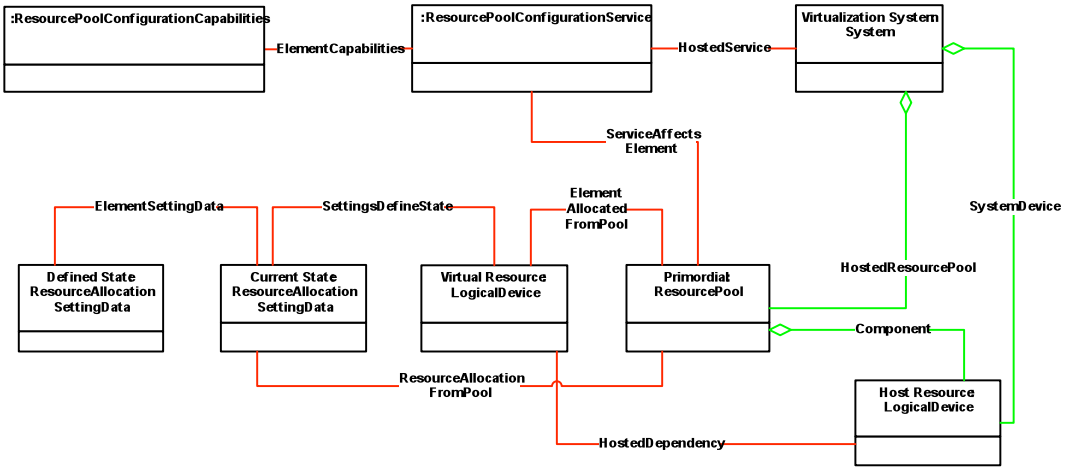


Figure 3 – Abstract Instance Diagram: Primordial Pool with Backed Resources

Figure 4 illustrates the use of the *Resource Allocation Profile* with a primordial pool that does not have backed resources used for virtualization. The resources are either synthetic (that is, no physical elements are backing them) or not modeled by the implementation.

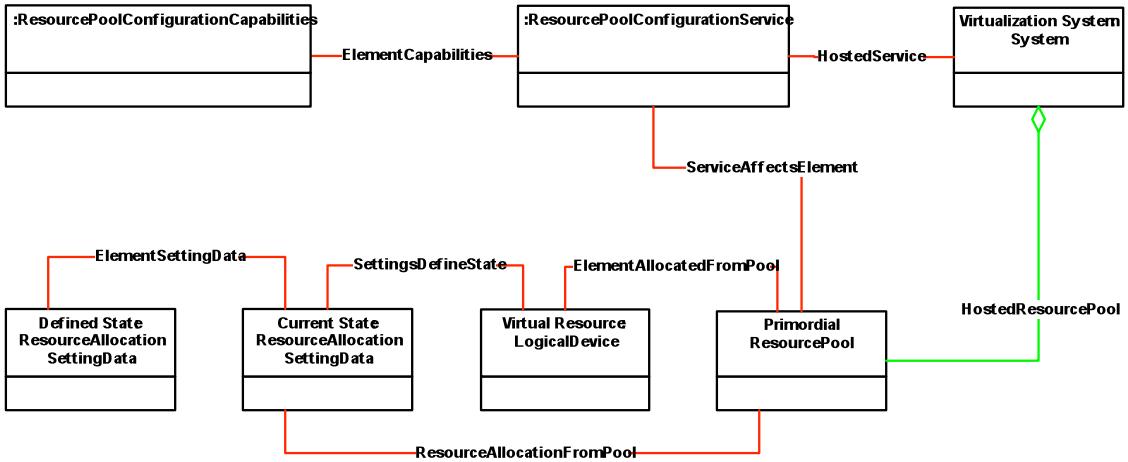


Figure 4 – Abstract Instance Diagram: Primordial Pool without Backed Resources

9.2 Resource Pool Hierarchy Diagram

Figure 5 shows a hierarchy of related resource pools in which host resources are shared. Child resource pools are allocated from a parent resource pool by using the same pattern as virtual resources. The host resources are members of the top-most or primordial resource pool. An instance of CIM_ResourceAllocationSettingData for a descendant resource pool records the way resources flow from the parent resource pool to the child resource pool. For example, if only weight is set, the child resource pool shares all resources with other child resource pools that have allocations scheduled based on the weight of the child resource pool.

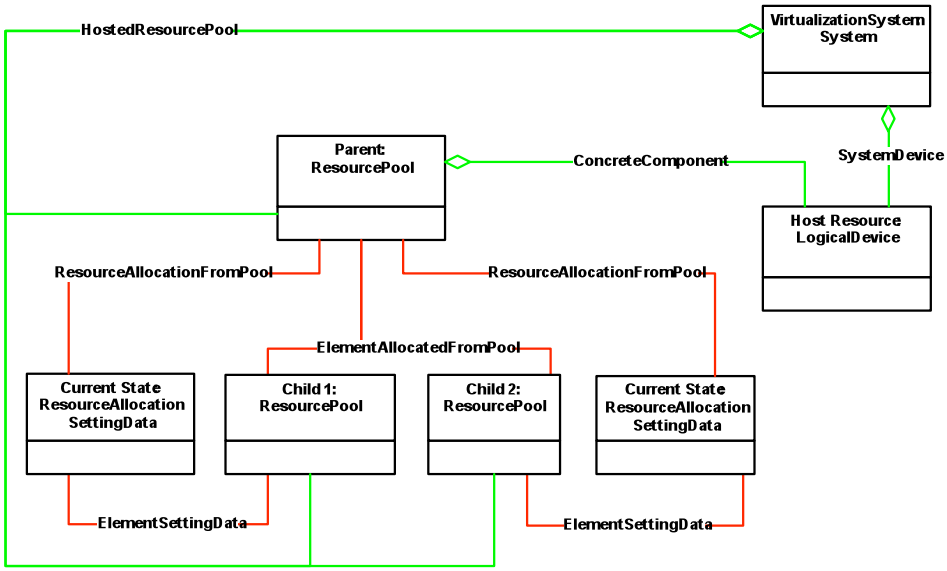


Figure 5 – Resource Pool Hierarchy Instance Diagram

9.3 Simple Resource Allocation Diagram

Figure 6 shows Simple Resource Allocation. Two instances of a subclass of CIM_LogicalDevice are aggregated into the CIM_ResourcePool instance through the CIM_ConcreteComponent association. This indicates that the components modeled by the device contribute resources into a pool from which the resources may be allocated.

CIM_LogicalDevice is shown to indicate that numerous different component types may be aggregated into the pool. rasd2 and rasd1 represent allocations from the pool for two resource consumers represented by Mse2 and Mse1. These allocations are indicated by the CIM_ResourceAllocationFromPool associations between rasd1 and rasd2 and the CIM_ResourcePool instance, and the CIM_ElementSettingData associations between the CIM_ResourceAllocationSettingData instances and the CIM_ManagedSystemElement instances.

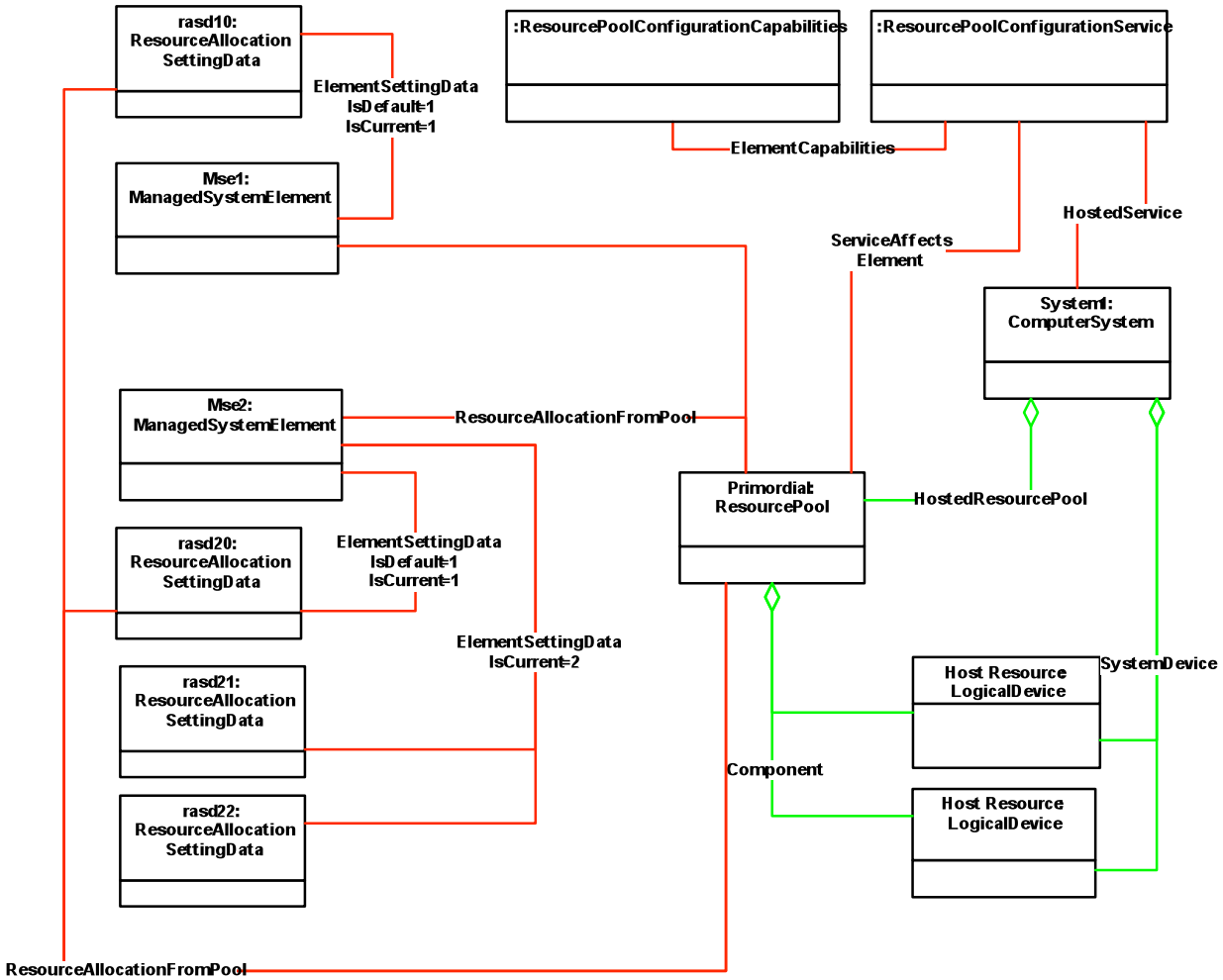


Figure 6 – Simple Resource Allocation

9.4 Determine Pool Type

A client may determine the type of resource provided by the resource pool by querying the ResourceType property in an instance of CIM_ResourcePool.

9.5 View Historical Use of Pool Resource by a Resource Consumer

The historical use of a resource by a particular resource consumer may be useful for reporting or billing purposes. It may also be useful for planning purposes in order to predict future use of the resource. A client may determine the historical use of a resource by a consumer as follows:

1) If device resource allocation is implemented, find all instances of CIM_LogicalDevice that are associated with the resource consumer of the type that corresponds to the resource type. Use each CIM_LogicalDevice instance as the subject instance in step 3) and sum the results.
2) If simple resource allocation is implemented, use the CIM_ManagedSystemElement instance that represents the resource consumer as the subject instance in step 3).
3) Find an instance of CIM_BaseMetricDefinition, as defined in 10.2.3, that is associated with the subject instance through the CIM_MetricDefForME association.
4) Find all instances of CIM_BaseMetricValue that are associated with the CIM_BaseMetricDefinition through the CIM_MetricInstance association where the CIM_BaseMetricValue.BreakdownValue property has the value of the PoolID property of the CIM_ResourcePool instance of interest.

9.6 View Historical Aggregate Use of a Pool Resource

The historical aggregate use of resources in the pool may be useful in predicting future resource requirements. A client may determine the historical aggregate use of a resource by a consumer as follows:

1) Find an instance of CIM_BaseMetricDefinition, as defined in 10.2.3, that is associated with the CIM_ResourcePool instance through the CIM_MetricDefForME association.
2) Find all instances of CIM_BaseMetricValue that are associated with the CIM_BaseMetricDefinition through the CIM_MetricInstance association where the CIM_BaseMetricValue.BreakdownValue property has the value of the PoolID property of the CIM_ResourcePool instance of interest.

9.7 Discover Host Resources

A client may find all the host resources of a specific type as follows:

1) Find instances of CIM_ResourcePool with the Primordial property set to TRUE and the ResourceType property set as desired.
2) Find all instances of CIM_ManagedSystemElement that are associated with the CIM_ResourcePool instances through the CIM_ConcreteComponent association.

9.8 Discover Supported Resource Types

A client may find all the resource types supported by the allocation platform as follows:

1) Enumerate resource pools and find primordial pool types.
2) Identify the unique ResourceType property values within the list of CIM_ResourcePool instances.

10 CIM Elements

Table 12 lists CIM elements that are specified or specialized for this profile. Each CIM element shall be implemented as described in Table 12. Clauses 7 (“Implementation”) and 8 (“Methods”) may impose additional requirements on these elements.

Table 12 – CIM Elements: Resource Allocation Profile

Element Name	Requirement	Description
Classes		
CIM_AffectedJobElement	Optional	See 10.1.
CIM_BaseMetricDefinition	Optional	See 10.2, 10.2.1, 10.2.2, and 10.2.3.
CIM_BaseMetricValue	Optional	See 10.3, 10.3.1, 10.3.2, and 10.3.3.
CIM_Component	Conditional	See 10.4.
CIM_ConcreteJob	Optional	See 10.5.
CIM_ElementAllocatedFromPool	Mandatory	See 10.6.
CIM_ElementCapabilities	Mandatory	See 10.7.
CIM_ElementSettingData	Mandatory	See 10.8.
CIM_HostedResourcePool	Mandatory	See 10.10.
CIM_HostedService	Mandatory	See 10.11.
CIM_LogicalDevice (virtual resource)	Mandatory	See 10.12.
CIM_MetricDefForME	Conditional	See 10.13.
CIM_MetricForME	Conditional	See 10.14.
CIM_ResourceAllocationFromPool	Optional	See 10.15.
CIM_ResourceAllocationSettingData	Conditional	See 10.16 and 10.17.
CIM_ResourcePool	Mandatory	See 10.18.
CIM_ResourcePoolConfigurationCapabilities	Mandatory	See 10.19.
CIM_ResourcePoolConfigurationService	Mandatory	See 10.20.
CIM_SettingsDefineState	Mandatory	See 10.21.
CIM_ServiceAffectsElement	Mandatory	See 10.22.
CIM_SystemDevice	Conditional	See 10.23.
CIM_HostedDependency	Optional	See 10.9.
Indications		
None defined in this profile		

10.1 CIM_AffectedJobElement

If long-running jobs are supported, this association provides a reference to the affected element. For example, if a new CIM_ResourcePool instance is created and a CIM_ConcreteJob instance is returned, after that CIM_ConcreteJob instance indicates that the create operation has completed the CIM_AffectedJobElement association may be used to locate the resulting CIM_ResourcePool instance. Table 13 defines the requirements for elements of this class.

Table 13 – Class: CIM_AffectedJobElement

Elements	Requirement	Notes
AffectedElement	Mandatory	The affected element (for example, the CIM_ResourcePool) Cardinality 1
AffectingElement	Mandatory	The CIM_ConcreteJob Cardinality 1

10.2 CIM_BaseMetricDefinition

CIM_BaseMetricDefinition defines metrics that are maintained for the resource pool and resource consumers. Table 14 defines the requirements for elements of this class.

Table 14 – Class: CIM_BaseMetricDefinition

Elements	Requirement	Notes
BreakdownDimensions	Mandatory	Matches ("CIM_ResourcePool.PoolID")
Calculatable	Mandatory	None
ChangeType	Mandatory	None
DataType	Mandatory	None
ElementName	Mandatory	Pattern (".+").
GatheringType	Mandatory	None
ID	Mandatory	**Key**
IsContinuous	Mandatory	None
TimeScope	Mandatory	None
TimeScope	Optional	None
Units	Mandatory	None

10.2.1 CIM_BaseMetricDefinition — Instantaneous Consumption

CIM_BaseMetricDefinition defines metrics that are maintained for the resource pool and resource consumers. Table 15 describes the requirements for using CIM_BaseMetricDefinition to define the metric for instantaneous consumption. These constraints are in addition to those specified in 10.2.

Table 15 – Class: CIM_BaseMetricDefinition — Instantaneous Consumption

Elements	Requirement	Notes
Calculatable	Mandatory	Matches 3 (Non-summable)
ChangeType	Mandatory	Matches 4 (Gauge)
DataType	Mandatory	Matches 13 (uint64)
ElementName	Mandatory	Pattern (".+")
TimeScope	Mandatory	Matches (Point)

10.2.2 CIM_BaseMetricDefinition — Interval Metrics

CIM_BaseMetricDefinition defines metrics that are maintained for the resource pool and resource consumers. Table 16 describes the requirements for using CIM_BaseMetricDefinition to define the metric for interval metrics. These constraints are in addition to those specified in 10.2.

Table 16 – Class: CIM_BaseMetricDefinition — Interval Metrics

Elements	Requirement	Notes
Calculatable	Mandatory	Matches 2 (Summable)
ChangeType	Mandatory	Matches 4 (Gauge)
DataType	Mandatory	Matches 13 (uint64)
TimeScope	Mandatory	Matches 3 (Interval)

10.2.3 CIM_BaseMetricDefinition — Aggregate Consumption

CIM_BaseMetricDefinition defines metrics that are maintained for the resource pool and resource consumers. Table 17 describes the requirements for using CIM_BaseMetricDefinition to define the metric for aggregate consumption. These constraints are in addition to those specified in 10.2.

Table 17 – Class: CIM_BaseMetricDefinition — Aggregate Consumption

Elements	Requirement	Notes
Calculatable	Mandatory	Matches 3 (Non-summable)
ChangeType	Mandatory	Matches 3 (Counter)
DataType	Mandatory	Matches 13 (uint64)
TimeScope	Mandatory	Matches 3 (Interval)

10.3 CIM_BaseMetricValue

CIM_BaseMetricValue conveys the actual defined data of a metric that has been maintained for a resource pool or resource consumer. Table 18 defines the requirements for elements of this class.

Table 18 – Class: CIM_BaseMetricValue

Elements	Requirement	Notes
MetricDefinitionID	Mandatory	None
MetricValue	Mandatory	None
Duration	Optional	None
TimeStamp	Optional	None
Volatile	Mandatory	None
InstanceID	Mandatory	**Key**
BreakdownDimension	Mandatory	Matches ("CIM_ResourcePool.PoolID")
BreakdownValue	Mandatory	Shall match the value of the CIM_ResourcePool.PoolID property for the pool from which the resource was consumed

10.3.1 CIM_BaseMetricValue — Instantaneous Consumption

CIM_BaseMetricValue reports a metric that is defined using CIM_BaseMetricDefinition. Table 19 describes the requirements for using CIM_BaseMetricValue to report the metric for instantaneous consumption. These constraints are in addition to those specified in 10.3.

Table 19 – Class: CIM_BaseMetricValue — Instantaneous Consumption

Elements	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

10.3.2 CIM_BaseMetricValue — Interval Metrics

CIM_BaseMetricValue reports a metric that is defined using CIM_BaseMetricDefinition. Table 20 defines the requirements for using CIM_BaseMetricValue to report the metric for interval metrics. These constraints are in addition to those specified in 10.3.

Table 20 – Class: CIM_BaseMetricValue — Interval Metrics

Elements	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

10.3.3 CIM_BaseMetricValue — Aggregate Consumption

CIM_BaseMetricValue reports a metric that is defined using CIM_BaseMetricDefinition. Table 21 defines the requirements for using CIM_BaseMetricValue to report the metric for aggregate consumption. These constraints are in addition to those specified in 10.3.

Table 21 – Class: CIM_BaseMetricValue — Aggregate Consumption

Elements	Requirement	Notes
Duration	Mandatory	None
Timestamp	Mandatory	None
Volatile	Mandatory	Matches TRUE

10.4 CIM_Component

CIM_Component associates a host resource with the resource pool. Table 22 defines the requirements for elements of this class.

Table 22 – Class: CIM_Component

Elements	Requirement	Notes
PartComponent	Mandatory	Shall be a reference to an instance of CIM_ManagedElement that represents a Host Resource Cardinality *
GroupComponent	Mandatory	Shall be a reference to an instance of CIM_ResourcePool Cardinality 0..1

10.5 CIM_ConcreteJob

CIM_ConcreteJob is used to manage the results of long-running operations to manage resource pools. Table 23 defines the requirements for elements of this class.

Table 23 – Class: CIM_ConcreteJob

Elements	Requirement	Notes
ElementName	Mandatory	(pattern ".*")
InstanceID	Mandatory	None
JobState	Mandatory	None
DeleteOnCompletion	Mandatory	Matches TRUE
ErrorCode	Mandatory	None
ErrorDescription	Mandatory	None
JobStatus	Mandatory	None
TimeBeforeRemoval	Mandatory	None

10.6 CIM_ElementAllocatedFromPool

CIM_ElementAllocatedFromPool is used to associate a CIM_LogicalElement that represents a virtual resource or child pool with the CIM_ResourcePool instance from which the resource was allocated. Table 24 defines the requirements for elements of this class.

Table 24 – Class: CIM_ElementAllocatedFromPool

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_ResourcePool Cardinality 1
Dependent	Mandatory	Shall be a reference to an instance of a subclass of CIM_LogicalElement that represents the virtual resource or child pool Cardinality *

10.7 CIM_ElementCapabilities

CIM_ElementCapabilities associates a resource pool configuration service to the capabilities instance that describes the methods supported by the service. Table 25 defines the requirements for elements of this class.

Table 25 – Class: CIM_ElementCapabilities

Elements	Requirement	Notes
Capabilities	Mandatory	Shall be a reference to an instance of CIM_ResourcePoolConfigurationCapabilities Cardinality 1
ManagedElement	Mandatory	Shall be a reference to an instance of CIM_ResourcePoolConfigurationService Cardinality 1..*

10.8 CIM_ElementSettingData

The CIM_ElementSettingData association shall be used to associate an instance of the CIM_SettingData class that represents a resource allocation as part of the resource allocation state with corresponding instances of the CIM_ResourceAllocationSettingData class that describe the same allocation element for the virtual resource in a different context, such as, for example, the resource allocation definition. Table 26 defines the requirements for elements of this class.

Table 26 – Class: CIM_ElementSettingData

Elements	Requirement	Notes
ManagedElement	Mandatory	Shall be a reference to an Allocation Target Cardinality *
SettingData	Mandatory	Shall be a reference to an instance of CIM_ResourceAllocationSettingData Cardinality 1..*

10.9 CIM_HostedDependency

CIM_HostedDependency associates a virtual resource with a specific host resource. Table 27 defines the requirements for elements of this class.

Table 27 – Class: CIM_HostedDependency

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_LogicalDevice that represents a Host Resource Cardinality 0..1
Dependent	Mandatory	Shall be a reference to an instance of CIM_LogicalDevice that represents a Virtual Resource Cardinality 1

10.10 CIM_HostedResourcePool

CIM_HostedResourcePool associates a resource pool with a hosting system. Table 28 defines the requirements for elements of this class.

Table 28 – Class: CIM_HostedResourcePool

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to the Host Instance Cardinality 1
Dependent	Mandatory	Shall be a reference to the Central Instance Cardinality 1..*

10.11 CIM_HostedService

CIM_HostedService associates a CIM_ResourcePoolConfigurationService with a host system. Table 29 defines the requirements for elements of this class.

Table 29 – Class: CIM_HostedService

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_System Cardinality 1
Dependent	Mandatory	Shall be a reference to an instance of CIM_ResourcePoolConfigurationService Cardinality *

10.12 CIM_LogicalDevice (Virtual Resource)

CIM_LogicalDevice is used to represent a virtual resource. Table 30 defines the requirements for elements of this class.

Table 30 – Class: CIM_LogicalDevice

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	**Key**
SystemName	Mandatory	**Key**
CreationClassName	Mandatory	**Key**
DeviceID	Mandatory	**Key**

10.13 CIM_MetricDefForME

CIM_MetricForME relates a metric to the managed element for which it was measured. Table 31 defines the requirements for elements of this class.

Table 31 – Class: CIM_MetricDefForME

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_ManagedElement Cardinality 1
Dependent	Mandatory	Shall be a reference to CIM_BaseMetricDefinition Cardinality *

10.14 CIM_MetricForME

CIM_MetricForME relates a metric to the managed element for which it was measured. Table 32 defines the requirements for elements of this class.

Table 32 – Class: CIM_MetricForME

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_ManagedElement Cardinality 1
Dependent	Mandatory	Shall be a reference to CIM_BaseMetricInstance Cardinality *

10.15 CIM_ResourceAllocationFromPool

CIM_ResourceAllocationFromPool is used to associate an instance of CIM_ResourceAllocationSettingData with the CIM_ResourcePool instance from which the resource was allocated. Table 33 defines the requirements for elements of this class.

Table 33 – Class: CIM_ResourceAllocationFromPool

Elements	Requirement	Notes
Antecedent	Mandatory	Shall be a reference to an instance of CIM_ResourcePool class that represents a resource pool. Cardinality 0..1
Dependent	Mandatory	Shall be a reference to an instance of CIM_ResourceAllocationSettingData. Cardinality *

10.16 CIM_ResourceAllocationSettingData (Resource Allocation)

An instance of class CIM_ResourceAllocationSettingData shall be used to represent a resource allocation. If a virtualized resource is allocated and its CIM_LogicalDevice subclass instance is present, an instance of this class shall also be present to reflect the current virtual resource allocation settings. Table 34 defines the requirements for elements of this class.

Table 34 – Class: CIM_ResourceAllocationSettingData (Current Settings)

Elements	Requirement	Notes
Address	Optional	None
AllocationUnits	Mandatory	None
AutomaticAllocation	Optional	None
AutomaticDeallocation	Optional	None
Connection	Optional	None
HostResource[ ]	Optional	None
InstanceID	Mandatory	Opaque
IsVirtualized	Optional	None
Limit	Optional	None
MappingBehavior	Conditional	See 7.2.4 and 7.2.5.
OtherResourceType	Optional	None
Parent	Optional	None
PoolID	Mandatory	None
Reservation	Optional	None
ResourceSubType	Optional	None
ResourceType	Mandatory	None
VirtualQuantity	Optional	None
Weight	Optional	None

10.17 CIM_ResourceAllocationSettingData (Resource Allocation Request)

An instance of the CIM_ResourceAllocationSettingData class shall be used to represent a resource allocation request. Implementations may choose to use one instance to reflect both defined and current settings and point to references within the CIM_ElementSettingData association and the CIM_SettingsDefineState association, respectively. Table 35 defines the requirements for elements of this class.

Table 35 – Class: CIM_ResourceAllocationSettingData (Defined Settings)

Elements	Requirement	Notes
Address	Optional	None
AllocationUnits	Mandatory	None
AutomaticAllocation	Optional	None
AutomaticDeallocation	Optional	None
Connection	Optional	None
HostResource[ ]	Optional	None
InstanceID	Mandatory	Opaque
IsVirtualized	Optional	None
Limit	Optional	None
MappingBehavior	Conditional	See 7.2.4 and 7.2.5.
OtherResourceType	Optional	None
Parent	Optional	None
PoolID	Optional	None
Reservation	Optional	None
ResourceSubType	Optional	None
ResourceType	Mandatory	None
VirtualQuantity	Optional	None
Weight	Optional	None

10.18 CIM_ResourcePool

One or more CIM_ResourcePool instances may exist on a system for any given CIM_ResourceType instance. Table 36 defines the requirements for elements of this class.

Table 36 – Class: CIM_ResourcePool

Elements	Requirement	Notes
InstanceID	Mandatory	Opaque
PoolID	Mandatory	Opaque
Primordial	Mandatory	See 7.1.2.
Capacity	Conditional	See 7.1.2 and 7.2.1.
Reserved	Optional	None
ResourceType	Mandatory	None
OtherResourceType	Optional	None
ResourceSubType	Optional	None
AllocationUnits	Conditional	Condition: Reserved or Capacity is implemented
ElementName	Optional	Pattern (".+")

10.19 CIM_ResourcePoolConfigurationCapabilities

All implementations shall implement this capabilities class, setting the supported properties to reflect the individual CIM_ResourcePoolConfigurationService methods supported by the implementation. Implementations of the individual service methods shall be either synchronous or asynchronous, but not both. Synchronous implementations may return quickly or slowly, and shall never return a Job. Asynchronous implementations shall always return quickly. If the operation is long running, the implementation shall return a Job to track the operation. Table 37 defines the requirements for elements of this class.

Table 37 – Class: CIM_ResourcePoolConfigurationCapabilities

Elements	Requirement	Notes
AsynchronousMethodsSupported	Mandatory	None
SynchronousMethodsSupported	Mandatory	None

10.20 CIM_ResourcePoolConfigurationService

The CIM_ResourcePoolConfigurationService provides for active management of Resource Pools. It allows jobs to be started for the creation and deletion of ResourcePools as well as addition and subtraction of host resources from ResourcePools. Table 38 defines the requirements for the CIM_ResourcePoolConfigurationService class.

Table 38 – Class: CIM_ResourcePoolConfigurationService

Elements	Requirement	Notes
SystemCreationClassName	Mandatory	None
CreationClassName	Mandatory	None
SystemName	Mandatory	None
Name	Mandatory	None
CreateChildResourcePool	Conditional	See 8.1.
AddResourcesToResourcePool	Conditional	See 8.2.
RemoveResourcesFromResourcePool	Conditional	See 8.4.
DeleteResourcePool	Conditional	See 8.2.

10.21 CIM_SettingsDefineState

CIM_SettingsDefineState associates an instance of CIM_LogicalDevice that represents a virtual resource and an instance of CIM_ResourceAllocationSettingData that represents the virtualization-specific state of a virtual resource. Table 39 contains the requirements for elements of this class.

Table 39 – Class: CIM_SettingsDefineState

Elements	Requirement	Notes
ManagedElement	Mandatory	**Key:** Reference to an instance of CIM_LogicalDevice that represents a virtual resource Cardinality 0..1
SettingData	Mandatory	**Key:** Reference to an instance of CIM_ResourceAllocationSettingData that represents the virtualization-specific state of a virtual resource Cardinality 0..1

10.22 CIM_ServiceAffectsElement

CIM_ServiceAffectsElement associates a CIM_ResourcePool with the service used to manage it. Table 40 defines the requirements for elements of this class.

Table 40 – Class: CIM_ServiceAffectsElement

Elements	Requirement	Notes
AffectedElement	Mandatory	Shall be a reference to an instance of CIM_ResourcePool Cardinality *
AffectingElement	Mandatory	Shall be a reference to an instance of CIM_ResourcePoolConfigurationService Cardinality 1

10.23 CIM_SystemDevice

CIM_SystemDevice associates a resource with the system to which it belongs. Table 41 defines the requirements for elements of this class.

Table 41 – Class: CIM_SystemDevice

Elements	Requirement	Notes
GroupComponent	Mandatory	Shall be a reference to an instance of CIM_System Cardinality 1
PartComponent	Mandatory	Shall be a reference to an instance of CIM_LogicalDevice that represents a Resource Cardinality *

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.1.0a	2007/10/16	Larry Lamers	Preliminary Standard
1.1.0b	2008/11/26	Larry Lamers	Preliminary Standard refresh
1.1.0c	2008/12/05	Larry Lamers	Comment resolution of WG ballot
1.1.0d	2009/01/09	Larry Lamers	Corrected error in simple allocation model
1.1.0e	2009/03/31	Larry Lamers	Removed 10.21 CIM_ResourcePoolSettingData, moved acknowledgements to Forward, Updated references,
1.1.0f	2009/03/31	Larry Lamers	Interim revision – had format issues
1.1.0g	2009/04/22	Larry Lamers	Removed the last 3 paragraphs of clause 7.6
1.1.0h	2009/04/30	Larry Lamers	Removed enumerate instances other fixes from WG discussion
1.1.0i	2009/05/28	Larry Lamers	Fixed clause 8 per TC directive
1.1.0j	2009/06/04	Larry Lamers	Removed lines 1015-1019 per MJ comment
1.1.0	2009/06/22		DMTF Standard Release
1.1.0	2009/06/25	Larry Lamers	Put in correct fig 1, address George's and Andy's comments